

05-12-04

AF 2123

Application No.: 09/707,194

Filed: 11/06/2000

Group Art Unit: 2173

11

6-15-04

B. J. H.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Regarding the application:

Title: Toolbar Manipulations

Priority: 11/06/2000

Number: 09/707,194

Examiner: Cao (Kevin) Nguyen

Appellant: Gary Odom

Art Unit: 2173

RECEIVED

MAY 13 2004

Technology Center 2100

REPLY BRIEF TO EXAMINER'S ANSWER

Pursuant to 37 CFR 1.193 (b) (1), enclosed in triplicate, herein is a reply to Examiner's Answer, dated May 2, 2004, and mailed May 4, 2004. 37 CFR 1.193 (b) (1) and MPEP 1208.03 fail to disclose acceptable formatting for a reply brief, leaving appellant with no guidance. As such, this reply addresses only discrepancies and outstanding differences between appellant's original appeal brief and examiner's answer, and does so using the form and numbering for which appeal briefs are filed. Appellant thanks Examiner and the Appeal Board in advance for graciously overlooking any minor procedural errors by appellant and in good faith dealing with the substantive issues. Appellant shall of course correct any deficiencies in reply as required.

(2) RELATED APPEALS AND INTERFERENCES

Appellant had stated in the original appeal brief: "There are no related appeals or interferences." Examiner's Answer appears to have overlooked aforementioned appellant appeal brief statement, claiming: "The brief does not contain a statement identifying the related appeals and interferences..." Though the point should be moot in light of Examiner acceptance in presuming no related appeals or interferences, appellant respectfully takes

exception to the portion of Examiner's Answer that "the Board, however, may exercise its discretions to require an explicit statement as to the existence of any related appeals and interferences", as no further statement by appellant on this issue need be forthcoming considering appellant's previous adherence to protocol.

(3) STATUS OF CLAIMS

Appellant had stated in the original appeal brief: "The rejection of claim 4 is not appealed." Appellant had indicated in the listing of claims in the original appeal brief that claim 4 was canceled.

Examiner's Answer stated: "The statement of the status of the claims contained in the brief is correct." In Examiner's Answer section (10), however, Examiner states that: "Claim 4, is not in fact canceled. Claim 4 should not have appeared in appendix."

Appellant apologizes for any mistake in procedure. Appellant was adhering to CFR 1.192 (c) as best as possible, even though CFR 1.192 (c) excuses "an applicant who is not represented by a registered practitioner" to conforming with CFR 1.192 (c). Certainly appellant was repeatedly clear in accepting rejection of claim 4 and attempting to cancel claim 4 in good faith. Appellant is left wondering how to properly cancel a claim.

(8) ARGUMENT IN REPLY TO EXAMINER'S ANSWER (10) GROUNDS OF REJECTION AND (11) RESPONSE TO ARGUMENT

Examiner's Answer makes no rebuttal to applicant's appeal brief arguing the irrelevance and the lack of anticipation of the following cited art to the claims presently under appeal: 1993.08 IBM TDB; 6,133,915 (Arcuri). Applicant therefore presumes that Examiner's reasons for rejection on those grounds have been overcome, and states nothing further regarding that prior art.

Examiner in Examiner's Answer persists that 5,644,737 (Tuniman) anticipates claims 5, 7-8, 10-11, 13, 15-20. Applicant therefore addresses Examiner's rejections on a claim by claim basis, conformant with CFR 1.192 (c) (8) (iii), to "specify the errors in the rejection

and why the rejected claims are patentable under 35 U.S.C. 102, including any specific limitations in the rejected claims which are not described in the prior art relied upon in the rejection”.

For reading convenience, each claim under appeal is recited, along with Examiner’s Answer, and specific description of relevant sections of Tuniman, to allow ready verification of appellant’s arguments.

Where appellant has not quoted in their entirety the sections of Tuniman disclosure upon which Examiner bases his rejections, appellant respectfully encourages that the Appeal Board read the cited disclosure sections to validate appellant’s characterization of Tuniman as accurate.

As more information unfolds during analysis of different claims, more confirming evidence for appellant’s previous conclusions emerges. As such, appellant respectfully requests that the Appeal Board suspend judgment on the merits of individual claims until this entire reply is read. In full understanding of what Tuniman declared and suggested, one becomes crystal clear that Tuniman failed to anticipate any of the claims on appeal.

The central issue, really the only issue, is whether Tuniman’s toolbars merge into a single toolbar when put together in a “stacked” toolbar, thus making a Tuniman toolbar a tool group in the confines of a stacked toolbar. This is the basis of Examiner’s rejections. If instead, as Appellant contends, that Tuniman maintained the integrity of the toolbars when confined within a stacked toolbar, then one must conclude that the claims under appeal should be allowed.

The primary pieces of evidence supporting the continued integrity of toolbars is their behavior within a stacked toolbar, and the configuring of a stacked toolbar. Figures 3-4 & 6-7 show toolbars in a stacked toolbar: only one toolbar is exposed and operational at a time. Other toolbars lie inactive and hidden, relegated to iconic/textual representation only, their minimal representation allowing selection and activation, but not immediate use. Tuniman at (10:59-61) is most clear: “a tab 164 labeled “Toolbars” can be selected to enable a user to choose the specific toolbars that are included in a stack”. Thus, for Tuniman, toolbars in a

stacked toolbar retain their integrity as separate entities, only being confined, not merged, by a stacking toolbar.

The definition of a toolbar tool group is germane to all claims under appeal save 13. Page two of the 09/707,194 disclosure defines a tool group as follows: "Tools 2 are typically functionally segregated by group dividers 5. The set of tools 2 between group dividers 5, or between one end of a toolbar 1 and a group divider 5 is referred to as a group 6 of tools 2."

One of the primary revelatory aspects of 09/707,194 is making tool groups functional units, as distinct from the toolbar in which they reside. Keeping this in mind helps both clarify the intended scope of the claims and their differences from the prior art.

MPEP 2143.03 – "To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)."

The dependent claims under appeal uniformly increase the limitations of the independent claim upon which they are based. In other words, a dependent claim has a lesser scope than its independent parent. Given this situation, failing to find anticipation in an independent claim necessarily means that a dependent claim is likewise not anticipated.

Claim 5

5. Software from at least one computer-readable medium directly altering the length of a tool group in a toolbar exclusive of editing any tools in said group or altering the length of said toolbar.

Examiner's Answer

The Tuniman '737 METHOD AND SYSTEM PROVIDING A COMPUTER FOR STACKING TOOLBARS IN A COMPUTER DISPLAY patent teaches the limitations recited in claim 5, Tuniman discloses software from at least one computer-readable

medium(s) directly altering the length of a tool group in a toolbar exclusive of editing any tools in said group or altering the length of said toolbar [..A stacked toolbar can also be docked on the left or right edges of the display screen or window to cover, it to a vertical toolbar, such as a toolbar ... see col. 7, lines 32-58].

At page 6, of the brief, appellant attempts to distinguish "altering the length of a tool group in a toolbar exclusive of editing any tools in said group or altering the length of said toolbar" from Tuniman's "plurality of toolbars that include graphic objects, which are arranged in a stack", arguing that Tuniman fails to anticipate "separating tool groups within a toolbar". However, what is actually recited in independent claims 5, 13, 15, 17 and 19-20 are how to directly altering the length of a tool group in a toolbar exclusive of editing any tools in said group or altering the length of said toolbar. This is anticipated as shown by the reference to Tuniman, which states "A stacked toolbar can also be docked on the left or right edges of the display screen or window to convert it to a vertical toolbar, such as a toolbar. When floating away from the screen or window edge, a stack toolbars can be readily resized by the user. It is contemplated that a dock toolbar might also be resizable. Although a floating stacked toolbar can also extend in a single row/column array of graphics objects if so sized by the user.." see col. 7, lines 32-67. However, toolbars holding a variety of selectable tools were created to offer users click access to actions and to enhance a user friendly interface.

Tuniman

Tuniman in (7:32-58) discloses how a "floating" horizontally arranged toolbar, depicted in Figures 3-5, may be "docked" to the top or bottom of the display screen or window in which it resides. Similarly, docking a toolbar to the left or right rearranges a toolbar to a vertically oriented single row of tools. Tuniman discloses how a stacked toolbar, showing a toolbar comprising multiple arbitrary rows of tools when floating, will become a single row of tools when docked.

Tuniman in paragraph (7:59-8:23) discloses how toolbars may be resized. Basically, a toolbar is a window holding tools that may be resized by a user to any rectangular

configuration, constrained to a size such that all the tools in the toolbar have room to appear. Tuniman explains: "Resizing is limited or controlled by a resizing algorithm to ensure that the area of the floating toolbar is sufficient to display all of the graphic objects on any toolbar in the stack. Thus, if the user decrease the length of the floating stacked toolbar, its width will increase automatically by an amount that accommodates the minimum area required by any toolbar in the stack." (8:17-23)

Appellant's Reply

Tuniman discloses resizing an entire toolbar in various ways.

The scope of Claim 5 explicitly excludes "altering the length of a toolbar", the very action which Tuniman discloses as the basis for Examiner's rejection.

As appellant has stated repeatedly, Tuniman fails to even mention or depict tool groups and their telltale dividers, and so in no way anticipates grouping tools within a toolbar, let alone manipulation of a tool group exclusive of toolbar manipulation, which is what claim 5 limits itself to.

Claim 5 clearly limits claim scope to a "altering the length of a tool group in a toolbar exclusive of ... altering the length of said toolbar". Tuniman discloses altering the length of a toolbar, but has no mention whatsoever of altering the length of a set of tools within a toolbar while not altering the length of a toolbar.

There is simply no evidence to support the assertion that Tuniman anticipated tool groups, as Tuniman never suggests it. Further evidence to support this view is provided later, as more quotations of Tuniman's are analyzed. Further evidence of Tuniman's conceptual orientation is provided by Tuniman's claims, all of which go to stacking toolbars, that is, manipulating two or more toolbars relationally within a superstructure (stacked toolbar), none of which go to partitioning a single tool bar into multiple groups. One cannot reasonably project by wishful thinking what was not stated.

Appellant suggests that U.S. patent 6,057,836 (Kavalam), which discloses manipulation of "composite" toolbars, is an antecedent basis of tool groups. Kavalam cited Tuniman.

09/707,194 cited Kavalam. If Tuniman could reasonably be construed as anticipating Kavalam in the concept of tool groups, then the claims of Kavalam would not have been granted. Ergo, the Examiner of Kavalam did not consider Tuniman to disclose tool groups.

If one considers that Tuniman confines toolbars together in a stacked toolbar superstructure, and toolbars are the cohesive unit to which Tuniman consistently refers, it is unreasonable to consider that Tuniman also anticipated tool groups when he makes no mention of a sub-grouping within a toolbar. One cannot project into a prior art reference what the author himself did not state.

Claim 7

7. Software according to claim 5 contracting the length of a tool group to hide at least one tool without a change in toolbar length.

Examiner's Answer

Regarding claim 7, Tuniman discloses contracting the length of a tool group to hide at least one tool without a change in tool bar length [..Resizing is limited or controlled by a resizing algorithm to ensure that the area of the floating toolbar is sufficient to display all of the graphic objects on any toolbar in the stack. Thus, if the user decrease the length of the floating stacked toolbar, its width will increase automatically by an amount that accommodates the minimum area required by any toolbar in the stack... see col. 8, lines 10-23].

Tuniman

Tuniman in (7:59-8:23) discloses how toolbars may be resized. Basically, a toolbar is a window holding tools that may be resized by a user to any rectangular configuration, constrained to a size such that all the tools in the toolbar have room to appear. As a user squeezes one dimension, such a width, a toolbar may need to lengthen to accommodate all tools. Tuniman explains: "Resizing is limited or controlled by a resizing algorithm to ensure that the area of the floating toolbar is sufficient to display all of the graphic objects on any

toolbar in the stack. Thus, if the user decrease the length of the floating stacked toolbar, its width will increase automatically by an amount that accommodates the minimum area required by any toolbar in the stack.” (8:17-23)

Appellant's Reply

Tuniman discloses resizing an entire toolbar or a “stacked” toolbar comprising multiple toolbars.

The scope of Claim 7 explicitly excludes “altering the length of a toolbar” by its dependency upon claim 5, and by its redundant statement of that limitation with “without a change in toolbar length”, the very action which Tuniman discloses, and the basis for Examiner’s rejection.

The point of claim 7 is “to hide at least one tool” by “contracting the length of a tool group”. Tuniman discloses hiding an entire toolbar in (10:34-52), or stacking toolbars such that a portion of one may be obscured (12:6-24), but never anticipates hiding a tool within a toolbar, let alone hiding a tool within a tool group, as claim 7 stipulates.

Claims 8 & 11

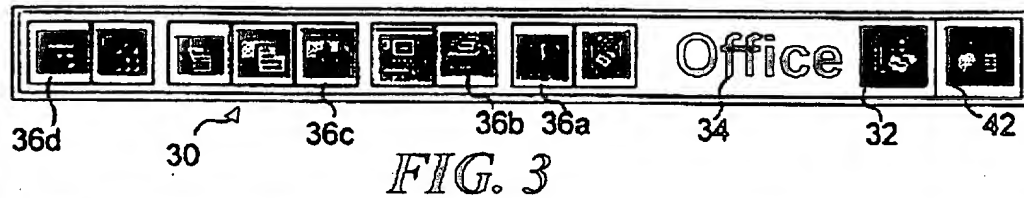
Claims 8 and 11 are covered here together owing to their similarity in claim limitations and Examiner’s singular rejection of both claims.

8. Software according to claim 7 by directly manipulating a tool group divider.

11. Software according to claim 10 by directly manipulating a tool group divider.

Examiner's Answer

Regarding claims 8 and 11, Tuniman discloses directly manipulating a tool group divider [see figures 3-4].

Tuniman

Tuniman discloses in Figures 3-4 in (6:21-52) the concept of stacked toolbars. Figure 3 shows the Office toolbar 30 fully displayed while the Desktop toolbar 38, represented only by its icon 34, is hidden. Conversely, in Figure 4, the Desktop toolbar 38 is in full glory, while the Office toolbar 30 is relegated to its icon 32.

Appellant's Reply

The scope of claim 8 is limited by its dependencies on claims 5 and 7. The arguments against rejection for claims 5 and 7 thereby apply to claim 8 as well.

The scope of claim 11 is limited by its dependency on claim 5. The arguments against rejection for claim 5 thereby apply to claim 11 as well.

Tuniman Figures 3-4 have no indication and the attendant disclosure makes no mention of a tool group or tool group divider, let alone tool group divider manipulation. In the face of those facts, Appellant cannot argue against Examiner's imagination.

Claim 10

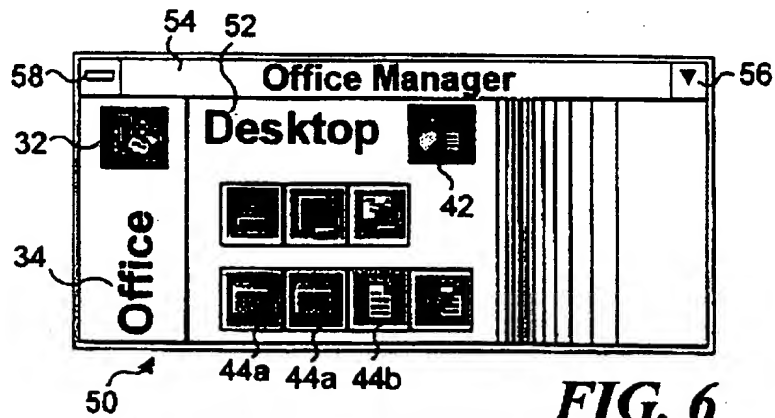
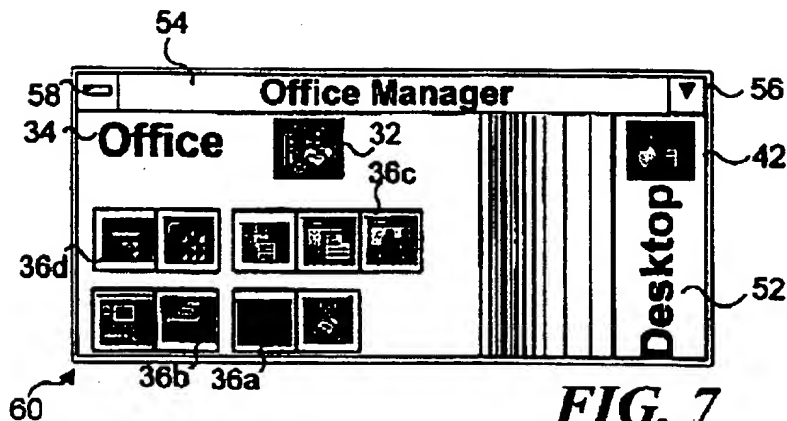
10. Software according to claim 5 expanding the length of a tool group to reveal at least one previously hidden tool without a change in toolbar length.

Examiner's Answer

Regarding claim 10, Tuniman discloses expanding the length of a tool group to reveal at least one previously hidden tool without a change in toolbar length [..the user can cause the graphic objects comprising the Office group to be fully disclosed as shown on a stacked toolbar col. 9, lines 13-33 and figure. 7].

Tuniman

Tuniman (9:13-33) is best understood by looking at Figures 6 & 7, which (9:13-33) describes.

**FIG. 6****FIG. 7**

Tuniman Figures 6 & 7 illustrate toggling between showing and hiding exemplary Office 50 and Desktop 52 toolbars by clicking on the target toolbar to display it. In Figure 6, the

Desktop toolbar 52 is displayed, and the Office toolbar 50 hidden. Note that icon 32 is a logo and control button, not a tool, and text label 34 is just that, nothing more.

Click on the hidden Office toolbar in Figure 6, and that causes the Office toolbar 50 to be displayed, as in Figure 7, at the expense of the Desktop toolbar 52

Appellant's Reply

The scope of claim 10 is limited by its dependency on claim 5. The arguments against rejection for claim 5 thereby apply to claim 10 as well.

Again, Examiner is construing Tuniman's toolbars as tool groups. As made most clear in analysis of Tuniman passage (10:59-61), cited by Examiner with regard to claim 13, Tuniman's toolbars are toolbars, not tool groups, even when they are confined together in a stacked toolbar.

In considering prior art anticipation, one must not view prior art in light of later developments, but as the one skilled in the art at the time of filing the prior art would understand the art.

Tuniman in Figures 6 & 7, and attendant disclosure, describes showing and hiding all tools of an entire toolbar. In showing the tools of a toolbar, the length of the toolbar changes, as clearly shown in Tuniman's figures. Claim 10 specifies showing a tool "without a change in toolbar length". One skilled in the art could not anticipate showing a tool without a change in toolbar length by reference to Tuniman.

Claim 13

13. Software from at least one computer-readable medium-directly merging two toolbars into one.

Examiner's Answer

Regarding claim 13, Tuniman discloses Software from at least one computer-readable medium directly merging two toolbars into one (see col. 10, lines 34-67).

At page 7, of the brief, Appellant argues that Tuniman does not teach or suggest "merge". However, the limitations as claimed are clearly anticipated as by the prior art that "the toolbar has been dragged into contact with the stacked toolbar snaps to the stack size and the graphic objects on the newly added toolbar are fully display." see col. 13, lines 40-45. This teaching, while not explicitly reciting the word "merge", as pointed out by appellants, clearly teach a merge function as the objects of one toolbar have been added an additional toolbar to the stack.

Tuniman

To allow the Appeal Board to conveniently review the material Examiner cites as damning to claim 13, appellant quotes the asserted relevant Tuniman passages in their entirety.

Tuniman writes in paragraph (10:34-52) how auto-hiding a stacked toolbar may work: "A further option provided for the stacked toolbars of FIGS. 3 through 5 is an auto-hide mode wherein the stacked toolbar is completely hidden except for a line a few pixels wide that extends vertically along the edge of the display screen. When the user jams the cursor against the edge of the display screen along which the fully hidden stacked toolbar is disposed, the stacked toolbar is again fully displayed, enabling the user to select any of the graphic objects on the selected toolbar or to choose a hidden toolbar so that its graphic objects are displayed. When the user moves the cursor off the fully displayed toolbar, it returns to the hidden position, i.e., appears to move off the screen or window. Optionally, the user can elect to have the fully hidden stacked toolbar "pop" into the fully displayed position, or can elect to have the hidden toolbar slide into the fully disclosed position, in an animated fashion. The auto-hide mode uses the least amount of display screen area, but the graphic objects on the selected toolbar are not available until the stacked toolbar is selectively displayed."

Tuniman describes in paragraph (10:53-61) view and configuration properties of a stacked toolbar, and as well how to add a tool to a selected toolbar: Various properties of a stacked toolbar can be selected in a properties dialog box 160 that is shown in FIGS. 10

through 12. In FIG. 10, a tab 162 is presently selected within dialog box 160, enabling the user to control the "View" properties of the stacked toolbar. A tab 163 labeled "Buttons" is selected in FIG. 11 to enable a user to configure the graphic objects used on a selected toolbar, and a tab 164 labeled "Toolbars" can be selected to enable a user to choose the specific toolbars that are included in a stack.

Tuniman elaborates in paragraph (10:62-11:14) on the view properties of a stacked toolbar: In the View properties, the user can select toolbar from a drop down box 165 and then set a Toolbar background color for the selected toolbar. The currently selected background color appears in a box 166. To change the color, the user selects a Change Color button 168, causing a screen with a color palette (not shown in the Figure) to be displayed, from which the user can select the desired background color. A check box 170 also enables the user to indicate whether a gradient fill should be added to the text on the stacked toolbars. The term "gradient fill" refers to a color scheme in which the background color gradually varies from a dark to a lighter intensity in the area where the text label for the toolbar is disposed (for example, on the right side of floating stacked toolbars 50 and 60 in FIGS. 6 and 7). A smooth characteristic for the background color is selected by placing a check in a check box 171. A check box 172 enables the user to selectively indicate that a Standard Toolbar Color should be used. The Standard Toolbar Color is determined by defaults for the desktop set in the graphic operating system.

Tuniman writes in paragraph (13:36-45) how a toolbar may be added to a stacked toolbar: "A negative response to decision block 242 leads to a decision block 248, which determines if the user wants to add an additional toolbar to the stack. If so, as indicated in a block 250, the user does so by clicking and dragging the background of a floating toolbar that is not part of the stack, so that as indicated in a block 252, the floating toolbar overlaps the stack. At this point, the toolbar that has been dragged into contact with the stacked toolbar snaps to the stack size and the graphic objects on the newly added toolbar are fully displayed. A negative response to decision block 248 returns to decision block 220."

Appellant's Reply

Tuniman writes in paragraph (10:34-52) how auto-hiding a stacked toolbar may work. Tuniman describes in paragraph (10:53-61) view and configuration properties of a stacked toolbar, and as well how to add a tool to a selected toolbar. Tuniman elaborates in paragraph (10:62-11:14) on the view properties of a stacked toolbar Tuniman describes in paragraph (13:36-45) how a toolbar is added to a stacked toolbar.

The most telling, relevant statement Tuniman makes in the Examiner-cited passages is at (10:59-61): “a tab 164 labeled "Toolbars" can be selected to enable a user to choose the specific toolbars that are included in a stack”. By allowing a user to see a list of toolbars that may or may not be in a stacked toolbar, Tuniman’s toolbars tab functionality clearly indicates that Tuniman’s toolbars maintain their integrity when added to a stacked toolbar; the toolbars are never merged or joined into a single functional unit as claim 13 stipulates.

Applicant disclosed the claimed process of merging toolbars as follow: “Figure 4 depicts two toolbars 1 arranged horizontally end-to-end. Depicted in Figure 5, toolbars 1 may be **merged (joined)** 11: a tail-end 21 toolbar 1 (in the depicted example, the menu toolbar 1m) may be joined 11 to a head-end 20 toolbar 1 (in the depicted example, the function toolbar 1f). The preferred embodiment to merge 11 toolbars 1 is by selecting the tail-end 21 toolbar handle 3 (toolbar 1m in the Figure) while pressing the ‘Ctl’ key, then dragging the mouse 107 pointer onto the back end of the head-end toolbar (1f in the Figure), then releasing the mouse 107 button; not much movement, distance-wise. Upon completion of a merge operation in the preferred embodiment, the toolbar handle 3 becomes a group divider 5.”

A patent applicant may act as his own lexicographer: please refer to MPEP 2106 and *Markman v. Westview Instruments*, 52 F.3d 967, 980, 34 USPQ2d 1321, 1330 (Fed. Cir.) (en banc), *aff'd*, U.S. , 116 S. Ct. 1384 (1996), among others.

For example, in defining a “tool group” in 09/707,194, as aforementioned, applicant stated a precise definition for a term, providing precise parameters of claim scope when the term is used as claim language.

As with the “tool group” definition example, most terms defined by patent applicants are nouns. Given the broad choices available, verbs less often need definitional refinement compared to nouns, and take their common definitional meaning; and so with the verb “merge” used in claim 13.

Merriam-Webster’s Third Unabridged Dictionary (MW 3UD) is widely regarded the most authoritative dictionary of the English language.

MW 3UD defines the transitive verb “merge” as: “to cause to combine, unite, or coalesce”

In the disclosure, applicant had used the transitive verb “join” alongside “merge”. This clarifies the intended definition of “merge” as being the meaning corresponding to both “merge” and “join”.

MW 3UD defines as the most common meaning of the transitive verb “join” as: “to put or bring together and fasten, connect, or relate so as to form a single unit, a whole, or a continuity”.

The clear intent of the 09/707,194 disclosure and claim 13 is that two toolbars become as one operational unit, indistinguishable from one another, forming a single toolbar from what had previously been two.

Tuniman never suggests or anticipates merging toolbars. Examiner concedes Tuniman never used the word “merge”.

Tuniman’s stacked toolbar may be most accurately defined as “confining” multiple toolbars together. MW 3UD defines the transitive verb “confine” as: “to hold within bounds : restrain from exceeding boundaries”.

Tuniman confines multiple toolbars together in a “stacked” toolbar. One need only look at Figures 3-4 and 6-7, shown above, to realize that only one of the toolbars in a stacked toolbar is fully visible and operational. The toolbars are never unified: toolbars in a stacked toolbar are shoved together into a confined space. The description Tuniman uses to explain Figures 3-4 & 6-7 bear this interpretation out. Tuniman’s description of the “Toolbars” tab in

(10:59-61) confirms the reasonable conclusion that Tuniman never suggested merging toolbars.

Claim 15

15. Software from at least one computer-readable medium automatically rearranging at least one tool based upon relative usage frequency of tools within a toolbar group.

Examiner's Answer

Regarding claim 15, Tuniman discloses automatically rearranging at least one tool based upon relative usage frequency of tools within a toolbar group [..a user will arrange a number of frequently used applications in different groups based on subject matter or common task relationship.; see col. 1, lines 30-37.]

Tuniman

In the passage cited in Examiner's Answer, Tuniman is prefacing the claimed invention by describing prior art, specifically Microsoft Windows® 3.1 Program Manager, and a related product called SideBar, as grouping application icons.

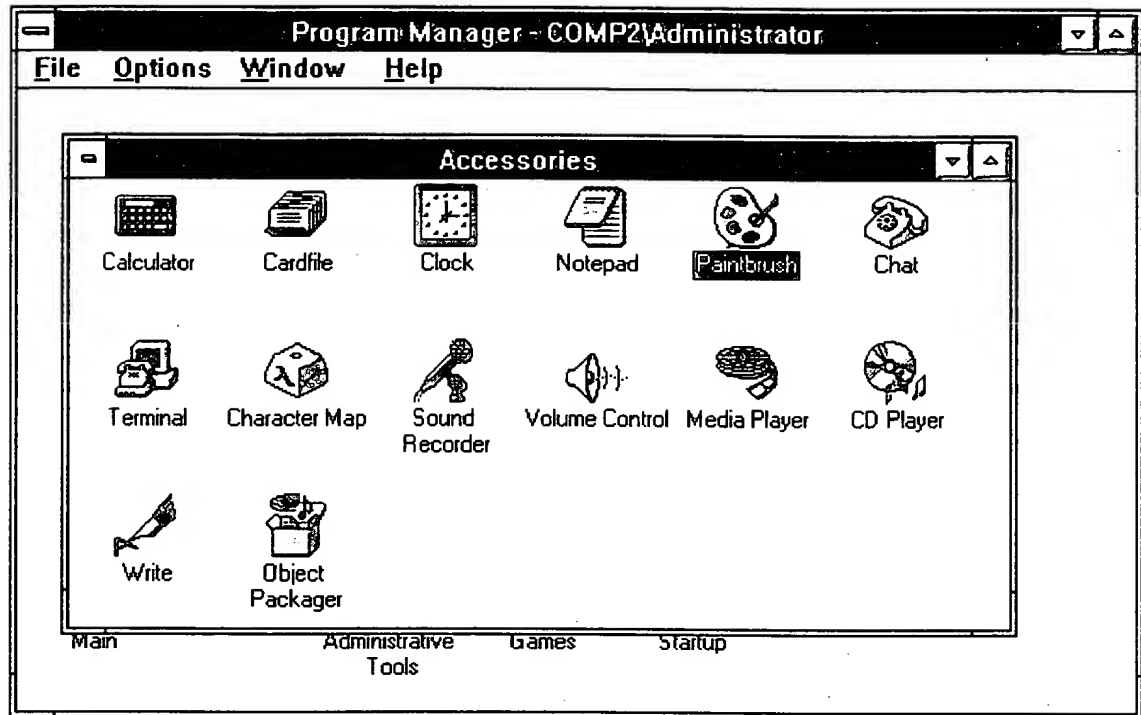
"Typically, a user will arrange a number of frequently used applications in different groups based on subject matter or common task relationship. Within a group, each application is represented by a graphic icon. In certain desktop shells, the graphic icons representing different applications are arranged in panels of buttons that can be configured in different rectangular shapes by dragging on a side of the panel with the cursor using the pointing device." (1:30-37)

Appellant's Reply

Tuniman's context is significant, and his prior art reference specific.

The application icons in Microsoft Windows® 3.1 Program Manager could be grouped, but did not constitute a toolbar. The Program Manager allowed a number of program groups,

each of which might contain a plurality of program items which, when double-clicked, would launch the application specified by the properties of the program item. For clarity in understanding, the figure below is a screen shot of Windows NT 3.1 Program Manager, with the Accessories window (not toolbar) frontmost.



Tuniman writes: "Typically, a user will arrange a number of frequently used applications in different groups based on subject matter or common task relationship."

First, Windows® 3.1 Program Manager groups did not comprise a toolbar, nor did a Program Manager group comprise a tool group. Second, Tuniman's remark is that a user may arrange the applications in groups, not that the software is capable, as claimed in claim 15, of "automatically rearranging at least one tool based upon relative usage frequency of tools". To equate a computer user's manual grouping of application icons in a window to a software facility "automatically rearranging tools with a tool group of a toolbar based upon relative usage frequency" is, respectfully, disingenuous.

Appellant recalls a product for Windows® 3.1 called SideBar, published by Quarterdeck Office Systems Inc., that provided the feature Tuniman describes: “the graphic icons representing different applications are arranged in panels of buttons that can be configured in different rectangular shapes by dragging on a side of the panel”.

Some quotations from Government Computer News, Dec 12, 1994 v13 n26 p37(1) in an article titled, “SideBar gives you a foretaste of what's ahead in Win95”, provide a general explanation of SideBar.

“SideBar's toolbar organizes directories and Windows® applications into folders that can be given long names and nested.” “It organizes directories and Windows program groups into folders, placed in a toolbar called, naturally, the sidebar. You can give your folders long file names and nest them.” “SideBar also lets you leave “shadows,” handy icons that launch your most-used applications. Associations, the links that tell Windows to launch an application when you click on its data file, are easier to make.”

The “shadow” icons referred to are what is now commonly known as “shortcuts”.

An understanding of the prior art to which Tuniman refers in his background reveals a failure in anticipating tool groups within toolbars, let alone automatic rearrangement of tools within a group based upon relative usage frequency. One skilled in the art at the time Tuniman's disclosure was filed could not possibly anticipate claim 15 by reading the passage Tuniman wrote, nor by knowing the software products Windows® 3.1 and SideBar.

The arguments presented in defense of claim 5 with regard to Tuniman failing to anticipate tool groups apply to claim 15 as well.

Tuniman makes no mention of monitoring tool usage. Without that, it is impossible to consider Tunimin as anticipating claim 15.

Claim 16

16. Software according to claim 15 preventing at least one tool from being rearranged.

Examiner's Answer

Regarding claims 16-18, Tuniman discloses preventing at least one tool from being rearranged; and automatically rearranging at least one group of a tools on a toolbar based upon aggregate usage frequency of tools within a tool group compared to another group (see col. 1, lines 38-53 and figures 6-80).

Tuniman

Tuniman writes in paragraph (1:38-53): "Graphic icons are also commonly used within applications to represent different tools, controls, commands, macros, or procedures. For example, Microsoft Corporation's Word for Windows includes several toolbars, each comprising a plurality of different graphic icon buttons that can be selected to carry out various tasks in the word processing system. The user can customize a toolbar or build a new one by adding or deleting graphic icon buttons. The appearance of any graphic icon button can be altered by editing the bitmap that appears on it in an icon editing utility. Toolbars are thus well known in the art and frequently used because they enable a user to immediately access tools within applications, just as the panels of graphic icon buttons that represent different applications or programs enable those applications to be immediately selected on the desktop of the graphic user interface."

Appellant's Reply

Tuniman describes prior art toolbars in (1:38-53), mentioning the ability to edit tools in a toolbar. Tuniman also mentions in the sentence "panels of graphic icon buttons" the concept embodied in SideBar, described above.

Claim 16 is dependent upon claim 15, and so the points made with regard to claim 15 not being anticipated by Tuniman apply as well to claim 16.

More specifically to claim 16, Tuniman makes no reference whatsoever to the concept of software automatically, without user intervention, during the process of rearranging tools

based on usage frequency, excluding a tool from the rearrangement process. No passage in Tuniman remotely suggests claim 16 limitations.

Claims 17-18

Claims 17 and 18 are covered here together owing to their similarity in claim limitations, claim 18 dependent upon 17 and with a single additional limitation, and Examiner's singular rejection of both claims.

17. Software from at least one computer-readable medium-automatically rearranging at least one group of a tools on a toolbar based upon aggregate usage frequency of tools within a tool group compared to another group.

18. Software according to claim 17 preventing at least one group from being rearranged.

Examiner's Answer

Regarding claims 16-18, Tuniman discloses preventing at least one tool from being rearranged; and automatically rearranging at least one group of a tools on a toolbar based upon aggregate usage frequency of tools within a tool group compared to another group (see col. 1, lines 38-53 and figures 6-80).

Tuniman

Tuniman writes in paragraph (1:38-53): "Graphic icons are also commonly used within applications to represent different tools, controls, commands, macros, or procedures. For example, Microsoft Corporation's Word for Windows includes several toolbars, each comprising a plurality of different graphic icon buttons that can be selected to carry out various tasks in the word processing system. The user can customize a toolbar or build a new one by adding or deleting graphic icon buttons. The appearance of any graphic icon button can be altered by editing the bitmap that appears on it in an icon editing utility. Toolbars are thus well known in the art and frequently used because they enable a user to immediately access tools within applications, just as the panels of graphic icon buttons that represent

different applications or programs enable those applications to be immediately selected on the desktop of the graphic user interface.”

Appellant's Reply

Tuniman describes prior art toolbars in (1:38-53), mentioning the ability to edit tools in a toolbar. Tuniman also refers in the sentence “panels of graphic icon buttons” to the concept embodied in SideBar, described above.

Claim 17 is readily understood to be a variant of claim 15. As such, the facts demonstrating Tuniman’s irrelevance in anticipation of claim 15 apply as well to claim 17. Particularly relevant is Tuniman’s failure to mention the concept of monitoring tool usage.

Tuniman never mentions anything related to software automatically acting upon comparative evaluation of tool usage between tool groups. No logical connection may be reasonably made.

Claim 18, as a dependent extension of claim 17 with an additional limitation of preventing at least one tool’s movement, relies for its statement of innovation upon the arguments marshaled to support claims 5, 15, 16, and 17. Claim 16 is relevant because claim 18 claims the same prevention of tool movement that claim 16 stipulates.

Claim 19

19. Software from at least one computer-readable medium directly selecting and moving a group of tools within a toolbar.

Examiner's Answer

Regarding claim 19, Tuniman discloses Software from at least one computer-readable medium directly selecting and moving a group of tools within a toolbar [..to drag the graphic object that is selected onto a toolbar comprising the stack, thereby adding the graphic object to the group of graphic objects within the toolbar; see col. 17, lines 12-17].

Tuniman

Examiner's reply is referring to 5,644,737 claim 18, which is dependent upon claim 1.

1. A method for providing access to a plurality of graphic objects on a computer display, comprising the steps of:

(a) organizing the plurality of graphic objects into a plurality of generally quadrilaterally shaped toolbars, each toolbar comprising a group of associated graphic objects organized in an array;

(b) creating a stack with the plurality of toolbars on the computer display, so that any selected toolbar is fully visible and hides a substantial portion of any non-selected toolbar from among the plurality of toolbars; a graphic object in any selected toolbar that is fully visible to a user on the computer display being directly selectable by the user to activate said graphic object; and

(c) enabling the user to choose any non-selected toolbar from among the plurality of toolbars of graphic objects, to make the non-selected toolbar that is thus chosen by the user become a selected toolbar that is fully visible to the user on the computer display, causing a previously selected toolbar to become a non-selected toolbar that is no longer fully visible, a substantial portion of the previously selected toolbar being substantially hidden by the toolbar just chosen by the user.

18. The method of claim 1, further comprising the step of enabling the user to use a pointing device to select a graphic object appearing on the computer display outside of the stack; and, to drag the graphic object that is selected onto a toolbar comprising the stack, thereby adding the graphic object to the group of graphic objects within the toolbar.

Appellant's Reply

For Tuniman, a toolbar comprises, as stated in claim 18, "the group of graphic objects within the toolbar". Tuniman does not divide a toolbar into separate sub-groups, that is, tool groups; there is only "the group within the toolbar".

So, Tuniman claims the concept of dragging and dropping a tool into a toolbar within the context of a stack of toolbars. As Tuniman claim 1 makes clear, the drag-and-drop of claim 18 may only occur to the currently selected toolbar in the toolbar stack, because, as claim 1 explicitly lays claim to, the other toolbars in the stack are hidden. This is hardly the concept of tool groups within a single toolbar, where all tool groups are displayed when the toolbar itself is visible. One could only reasonably conclude that Tuniman was referring to separate toolbars within a toolbar superstructure ("stacked" toolbar), not tool groups within a toolbar.

Tuniman claim 18 is for a single tool. Claim 19 under appeal is for "a group of tools within a toolbar". Tuniman never suggests being able to select or move "a group of tools within a toolbar" as stipulated in claim 19.

Claim 20

20. Software from at least one computer-readable medium directly merging a group of tools on a toolbar with another group of tools.

Examiner's Answer

Regarding claim 20, Tuniman discloses Software from at least one computer-readable medium directly merging a group of tools on a toolbar with another group of tools [...for adding and removing a selected graphic object respectively to and from the groups of graphic objects comprising the toolbars in the stack, by dragging and dropping the selected graphic object.; see col. 18, lines 43-48.]

Tuniman

Exmainer is referring to 5,644,737 claim 32, which is dependent upon claim 25.

25. A graphic operating system that is implemented on a computer, said graphic operating system including graphic objects that appear on a computer display, comprising:

(a) means for organizing the plurality of graphic objects into a plurality of generally quadrilaterally shaped toolbars, each toolbar comprising a group of associated graphic objects organized in an array;

(b) means for creating a stack of the toolbars on the computer display so that any selected toolbar substantially hides a substantial portion of any non-selected toolbar in the stack; said group of graphic objects in any selected toolbar being fully visible on the computer display to a user so that a graphic object within said group is directly selectable and activatable by the user; and

(c) means for enabling the user to choose a non-selected toolbar to become a selected toolbar, including means for causing the non-selected toolbar thus chosen to become fully visible so that the graphic objects comprising it are visible to the user on the computer display, and so that a substantial portion of a previously selected toolbar is hidden by the toolbar that was just chosen by the user, said graphic operating system thereby reducing an area of the computer display required for displaying the groups of graphic objects comprising the toolbars in the stack.

32. The graphic operating system of claim 25, further comprising means for adding and removing a specific toolbar respectively to and from the plurality of toolbars in the stack, by dragging and dropping the specific toolbar.

Appellant's Reply

Tuniman claim 32 adds to claim 25 dragging and dropping a toolbar into or out of a toolbar stack.

Tuniman nowhere advances or displays the concept of tool groups. As repeatedly demonstrated herein, the stacked toolbar Tuniman disclosed comprised individual toolbars, not tool groups. Furthermore, as claim 25 lucidly states, Tuniman's thinking is that, one presumes to avoid clutter and confusion, only one toolbar at a time is visible and selected: "any selected toolbar substantially hides a substantial portion of any non-selected toolbar in the stack". The concept of a stacked toolbar comprising tool groups simply never occurs to

Tuniman, as made clear by the fact he only wants one toolbar in a stacked toolbar visible at a time, let alone immediately functional.

Because Tuniman does not make any suggestion that toolbars themselves may be sub-grouped, thus failing to anticipate tool groups, and possible manipulations of tool groups as a separate entity from the toolbar in which they reside, Tuniman fails to anticipate the concept of allowing a user through direct manipulation merging groups of tools.

Appellant respectfully traverses rejection of appealed claims 5, 7-8, 10-11, 13, 15-20 with regard to the prior art, and respectfully submits that the appealed claims should be allowed. Considering what Appellant feels is obvious lack of anticipation of the claims by Tuniman, in light of Examiner's persistent failure to carefully and objectively distinguish prior art from the claim limitations, Appellant requests an appropriate extension for having to suffer undue delay in allowance.

Appellant thanks the Appeal Board for its consideration.

Respectfully submitted,



Gary Odom

15505 SW Bulrush Lane, Tigard, OR 97223

telephone: (503) 524-8371

fax: (775) 632-5551

date: May 10, 2004